ELL8299/AIL861/ELL881

Advances in Large Language Models

QUIZ 2 ANSWER KEY

Semester I, 2025-26

Answer Key — solutions are written in the spaces provided.

Total Marks: 40 **Time: 60 Minutes**

Part A: True or False ($5 \times 1 = 5$ Marks)

State whether the following statements are True or False. No justification is needed.

1. A2A protocol enables communication only between a user and a single local agent, with no support for remote agents.

   Answer:
   False. A2A supports multi-agent communication, including remote agents, so it is not limited to a single local agent.

2. Retrieval-Augmented Generation (RAG) is guaranteed to overcome the **hallucination** problem of standard parametric LLMs.

   Answer:
   False. Models can still hallucinate even after retrieving context from documents.

3. The "Toolformer" model relies on a large, human-annotated dataset of (instruction, API call) pairs to learn how to use tools.

   Answer:
   False. Toolformer teaches itself to use tools. It does this by sampling potential API calls, executing them, and then filtering to keep only the successful calls that help the model predict the next tokens better. It then finetunes itself on this self-generated data.

4. "Chain-of-Thought" (CoT) prompting is a finetuning technique that updates an LLM's weights to improve its reasoning abilities.

   Answer:
   False. CoT prompting is an inference-time technique. It guides the model to produce reasoning steps by providing few-shot examples (or the phrase "Let's think step by step") in the prompt, without changing the model's weights.

5. The "ImageBind" model's key finding is that a model must be trained on all possible modality pairs (e.g., audio-text, text-depth, audio-IMU) to create a single joint embedding space.

   Answer:
   False. The key finding is the opposite: "Image-paired data is sufficient to bind the modalities together." By aligning all other modalities (like audio, depth, IMU) to a common image embedding space, they become aligned with each other by proxy, without needing direct cross-training.

Part B: Short Answer Questions ($5 \times 2 = 10$ Marks)

Answer the following in 2-3 sentences each.

1. Explain with an example how a Large Language Model can use a calculator to perform numerical operations accurately.

   Answer:
   An LLM can learn to use a calculator by being trained on annotated examples where expressions like `<<20+10>>` trigger an external tool. During inference, the model detects the annotation, sends the expression to the calculator, receives the exact result (30), and inserts it back as `<<20+10=30>>`. This training teaches the model when to invoke the calculator and how to integrate the answer accurately.

2. Briefly describe the difference between **Sparse Retrieval** (e.g., BM25) and **Dense Retrieval** (e.g., DPR) in the context of Retrieval-based LMs.

   Answer:
   Sparse Retrieval (like BM25) works on exact keyword matching (lexical similarity). It represents documents as very large, sparse vectors (based on word counts/TF-IDF). **Dense Retrieval** (like DPR) works on semantic meaning. It uses a neural encoder (e.g., BERT) to map documents to smaller, dense vectors where similar meanings are close in the embedding space.

3. What is the key limitation of the basic "Self-Taught Reasoner" (STaR) training loop, and how does the "rationalization" step (using hints) address it?

   Answer:
   The key limitation is that the model is only trained on examples it can already solve correctly, so it gets no training signal from its failures. The "rationalization" step addresses this by providing the correct answer as a hint for failed problems, allowing the model to "reason backwards" and generate a correct reasoning trace to learn from.

4. Write the continuous-time equation of a State Space Model (SSM) used in sequence modeling, and define each term clearly.

   Answer:

   $$\dot{x}(t) = A\,x(t) + B\,u(t), \qquad y(t) = C\,x(t),$$

   where $x(t)$ is the state, $u(t)$ is the input, $y(t)$ is the output, and A, B, C are the state, input, and output matrices.

5. How does the **LLaVa** model architecturally combine a pre-trained Vision Encoder (like CLIP's) and a pre-trained LLM (like LLaMa)?

   Answer:
   LLaVa uses a simple **projection layer** (an MLP) to act as a bridge. This layer takes the output features from the frozen vision encoder and maps them into the word embedding space of the frozen LLM. This allows the LLM to "read" the visual features as if they were text tokens, enabling visual instruction tuning.

Part C: A bit Maths? ($1 \times 5 = 5$ Marks)

This question pertains to Knowledge Distillation losses.

1. The slides present several divergence functions used to measure the difference between a teacher's distribution (p) and a student's distribution (q).

   (a) Write the mathematical formula for **Forward KLD** and **Reverse KLD**. (2 marks)

   Answer:

   $$ForwardKLD : -KL(p||q) = \sum_t p(t) \log \frac{p(t)}{q(t)}$$

   $$ReverseKLD : -KL(q||p) = \sum_t q(t) \log \frac{q(t)}{p(t)}$$

   (b) What is the difference between black-box KD and white-box KD. Explain how a teacher's entropy affects the distillation loss with a fixed student. (2 marks)

   Answer:
   Black-box KD uses only the teacher's output probabilities, whereas white-box KD also transfers internal features such as hidden states or attention maps.
   The distillation loss is

   $$\mathrm{KL}(p_T \| q) = \mathbb{H}(p_T, q) - \mathbb{H}(p_T).$$

   With a fixed student, a higher-entropy teacher (larger $\mathbb{H}(p_T)$) reduces the KL loss, while a sharper teacher increases it.

   (c) According to the MiniLLM paper, why is Reverse KLD preferred over Forward KLD for distilling generative language models? (1 mark)

   Answer:
   Reverse KLD is preferred because it prevents the student model from "overestimating the low-probability regions of the teacher distribution". This forces the student to focus on matching the prominent modes (high-probability outputs) of the teacher, which is better for generation (higher precision).

Part D: Think before you Go! ($4 \times 5 = 20$ Marks)

Answer the following questions by synthesizing concepts from the lectures.

1. Describe the architecture of the RWKV model. Show how it computes the equivalent of attention without incurring the quadratic computational cost. (5)

Answer:

(a) Architecture of RWKV

RWKV replaces self-attention with two components: time mixing and channel mixing. Time mixing produces receptance, key, and value signals:

$$\tilde{x}_t = (1-\tau)x_t + \tau x_{t-1}, \qquad r_t = \sigma(W_r \tilde{x}_t), \quad k_t = W_k \tilde{x}_t, \quad v_t = W_v \tilde{x}_t.$$

Channel mixing applies an MLP-style transformation across feature dimensions.

(b) Attention-like computation without quadratic cost

RWKV maintains recurrent weighted states:

$$A_t = \delta_t A_{t-1} + e^{k_t} v_t, \qquad B_t = \delta_t B_{t-1} + e^{k_t},$$

$$\text{wkv}_t = \frac{A_t}{B_t + \varepsilon}, \qquad y_t = r_t \odot \text{wkv}_t.$$

Unrolling the recurrence yields a normalized weighted sum over past tokens:

$$\text{wkv}_t = \frac{\sum_{\tau \le t} \left(\prod_{s=\tau+1}^{t} \delta_s \right) e^{k_\tau} v_\tau}{\sum_{\tau \le t} \left(\prod_{s=\tau+1}^{t} \delta_s \right) e^{k_\tau}}.$$

This mimics prefix attention while avoiding the quadratic $O(T^2)$ cost of forming a full attention matrix; the update is computed in linear time $O(T)$.

2. Draw the retriever pipeline used in a Retrieval-Augmented Generation (RAG) model. Compare and contrast a **Retrieval-based LM** (like RAG) with an **Agentic Workflow** (like ReAct). (5)

Answer:

Diagram :- RAG part 1 ppt (page 37)

- **Purpose:** RAG's purpose is to **augment knowledge** for generation. It answers a query by finding relevant context and synthesizing it into a response. An Agentic Workflow (like ReAct) aims to **accomplish a task** by autonomously creating and executing a multi-step plan.
- **Method:** RAG performs a "retrieve-then-read" operation, typically once at the beginning. It finds relevant documents and conditions the LLM's single-pass answer on them. ReAct is an **iterative loop**: it generates a *Thought*, then an *Action* (like a tool call, e.g., 'Search[]'), receives an *Observation* (the tool's output), and uses that observation to generate the next thought.
- **Interactivity:** RAG is static; it retrieves information once and then generates. ReAct is dynamic and interactive; it can use tools multiple times, and the observation from one tool call (e.g., a search result) actively influences its next thought and subsequent action (e.g., a 'Lookup[]' or another search).

3. Large Language Models demonstrate reasoning abilities that allow them to chain logical steps and simulate human-like thought processes. With respect to reasoning in LLMs, answer the following:

   (a) Explain how the DeepSeek-R1-Zero model is trained to develop reasoning abilities. (2)

   (b) Explain the fundamental difference between Parallel Scaling and Sequential Scaling in the context of Scaling Test-Time Compute. (2)

   (c) While training LLMs to reason, it was observed that Qwen-3B quickly learned new reasoning skills, whereas LLaMA-70B showed little to no improvement, despite being a much bigger model. Discuss the possible reason behind this difference. (1)

Answer:
(a) DeepSeek-R1-Zero is trained using RL with verifiable rewards, where each reasoning trace is automatically scored for correctness. The model is optimized using GRPO, which incorporates both an output reward (accuracy) and a format reward (quality and structure of reasoning). This setup enables reliable improvement in multi-step reasoning without human annotations.
(b) Parallel scaling improves reasoning by running multiple independent samples simultaneously (e.g., majority voting), whereas sequential scaling allocates more compute to a single chain of thought (e.g., S1 forcing deeper reasoning).
(c) Qwen-3B improved because its training setup allowed its policy to effectively exploit the provided rewards, enabling rapid adaptation of reasoning behavior.

4. With respect to modern vision and vision-language models, answer the following.

   (a) How does the **Vision Transformer** process and represent images? (1)

   (b) What is the objective of **CLIP**? Draw the architecture or training pipeline of CLIP. (2)

   (c) Explain the role of the **Q-Former** in the BLIP-2. How does it act as a bridge? (2)

Answer:
(a) The Vision Transformer splits an image into fixed-size patches, flattens and linearly embeds them, adds positional encodings, and feeds the resulting patch tokens into a Transformer encoder. The final image representation is obtained from the `[CLS]` token or an average of patch embeddings.
(b) CLIP trains an image encoder and a text encoder jointly using a contrastive objective, aligning matching image–text pairs while pushing apart mismatched pairs. Its goal is to learn a shared embedding space where semantically similar images and texts are close together.
(c) The Q-Former is a "bridge" component that connects the **frozen image encoder** (e.g., ViT) to the **frozen LLM** (e.g., FlanT5). Its role is to efficiently **distill** the large set of visual features from the image encoder into a small, fixed number (e.g., 32) of "query" vectors. These vectors represent the most relevant visual information for the LLM. This small set of features is then fed as a "soft prompt" to the LLM, allowing it to "see" the image without requiring the massive LLM or vision model to be retrained.